AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1-15 (Canceled)

16. (Currently Amended) ~~The superscalar processor of claim 9,~~ A superscalar processor that, for a given instruction instance, performs, over plural execution cycles of the superscalar processor, instruction grouping for dispatch, including both intra-group and inter-group dependency checking, wherein the instruction grouping for dispatch takes the plural execution cycles to complete, wherein non-deterministic conditions are evaluated in a final stage of instruction grouping prior to dispatch.

17. (Previously Presented) A processor comprising:

plural functional units that execute instructions in respective numbers of processor cycles; and

grouping logic coupled to the functional units and pipelined to compute, over plural cycles, T, of the processor, a future state, S(t+T), of the processor based on a prior state, S(t), of the processor and based thereon to select a group of instructions from a program sequence thereof for dispatch to the functional units,

wherein the future state computing takes the plural cycles to complete.

18. (Previously Presented) The processor of claim 17, further comprising:

wherein intra-group dependency checking by the pipelined grouping logic spans two or more of the T cycles.

19. (Previously Presented) The processor of claim 17, further comprising:

wherein inter-group dependency checking is performed independently of intra-group dependency checking.

20. (Previously Presented) The processor of claim 17,

wherein intra-group dependencies are checked by the pipelined grouping logic beginning in a first of the T cycles; and

wherein non-deterministic dependency conditions are checked during a last of the T cycles.

21. (Previously Presented) The processor of claim 20,

wherein inter-group dependencies are checked independent of the intra-group dependencies.

22. (Previously Presented) The processor of claim 17,

wherein the grouping logic implements T stages of a pipeline of the processor.

23. (Previously Presented) The processor of claim 17,

wherein T is four.

24. (Previously Presented) The processor of claim 17,

wherein the functional units include at least one functional unit capable of receiving and completing an instruction for each of the processor cycles.

25. (Previously Presented) The processor of claim 17,

wherein the functional units include at least one functional unit requiring multiple of the processor cycles for receiving and completing an instruction.

26-29 (Canceled)

30. (Currently Amended) ~~the method of claim 26,~~ A method of operating a processor, the method comprising:

identifying successive groups of instructions for dispatch to respective ones of plural execution units of the processor;

performing, during plural pipelined execution cycles of the processor, dependency checking amongst instructions of a later one of the groups and between the instructions of the later group and instructions of a preceding one of the groups; and

dispatching instructions of the later group only after all instructions of the preceding group have been dispatched,

wherein the performed dependency checking takes the plural pipelined execution cycles to complete, and wherein non-deterministic conditions are evaluated in a final one of the pipelined execution cycles implemented by pipelined grouping logic.

31. (Currently Amended) [[The]] A method of grouping instructions for dispatch to execution units of a processor, the method comprising:

in a first cycle of processor execution, identifying plural candidate instructions for an instruction group;

in a subsequent cycle of processor execution, beginning intra-group dependency checking [[as]] amongst instructions of the instruction group;

in a cycle of processor execution prior to dispatch of any instruction from the instruction group, checking non-deterministic conditions; and

in a cycle of processor execution prior to the non-deterministic dependency condition checking, initiating inter-group dependency checking between instructions of the instruction group and instructions of one or more prior instruction groups,

wherein, for instruction instances of the instruction group, dependency checking, which includes the intra-group dependency checking and the inter-group dependency checking, takes plural of the cycles of processor execution to complete.

32. (Previously Presented) The method of claim 31, further comprising:

dispatching one or more instructions of the instruction group only after all instructions of the one or more prior instruction groups have been dispatched.

33. (Previously Presented) The method of claim 31,

wherein the non-deterministic condition checking is performed conservatively, based on an assumption of no change in condition.

34. (Previously Presented) The method of claim 31,

wherein the non-deterministic condition checking is performed aggressively, computing dispatch conditions for at least two alternatives including no change in condition and condition resolution; and

wherein a control signal is selective for a particular one of the computed dispatch conditions.

35-36. (Canceled)